

FIG. 2 Verifying Load Modules

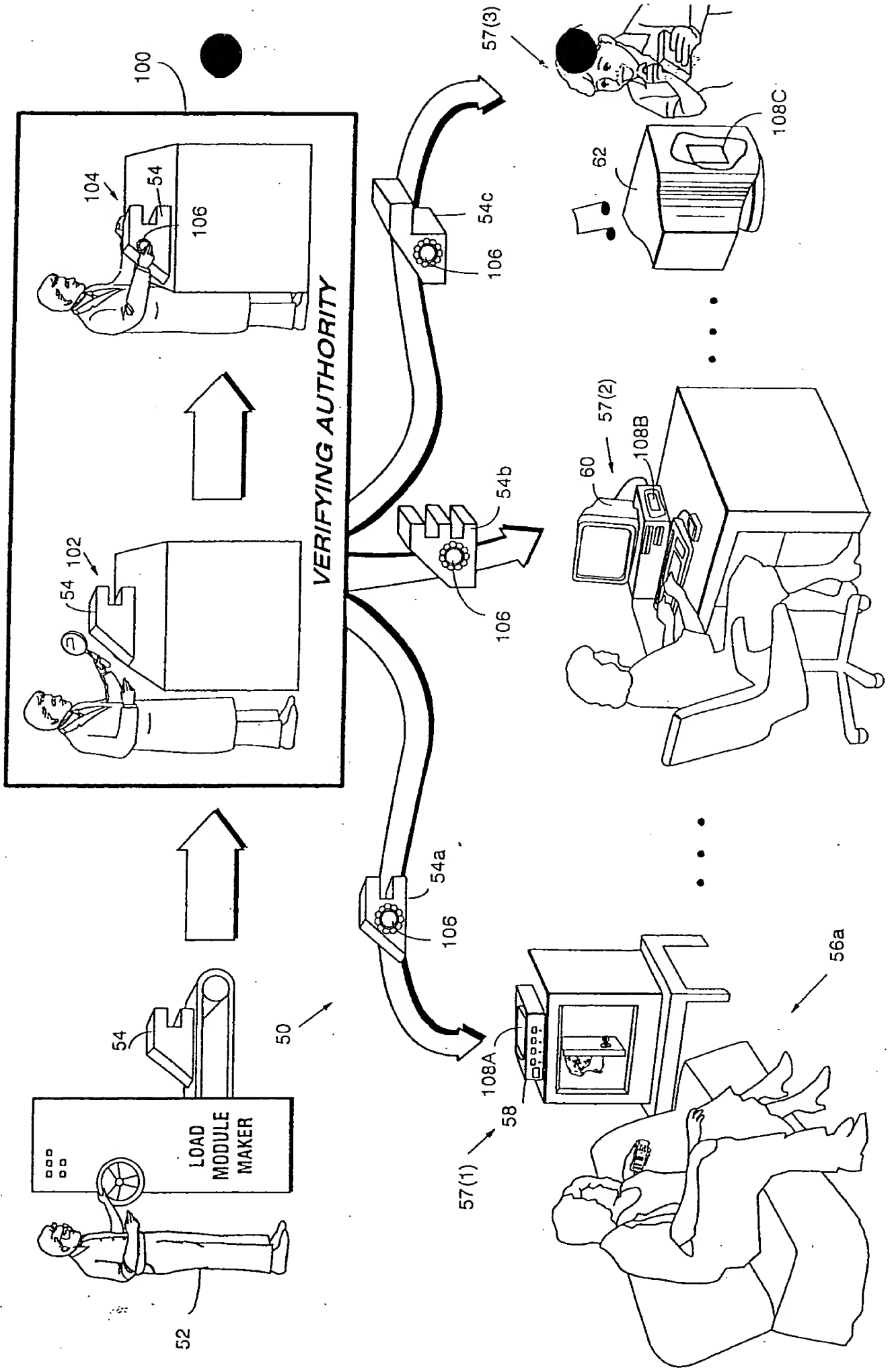
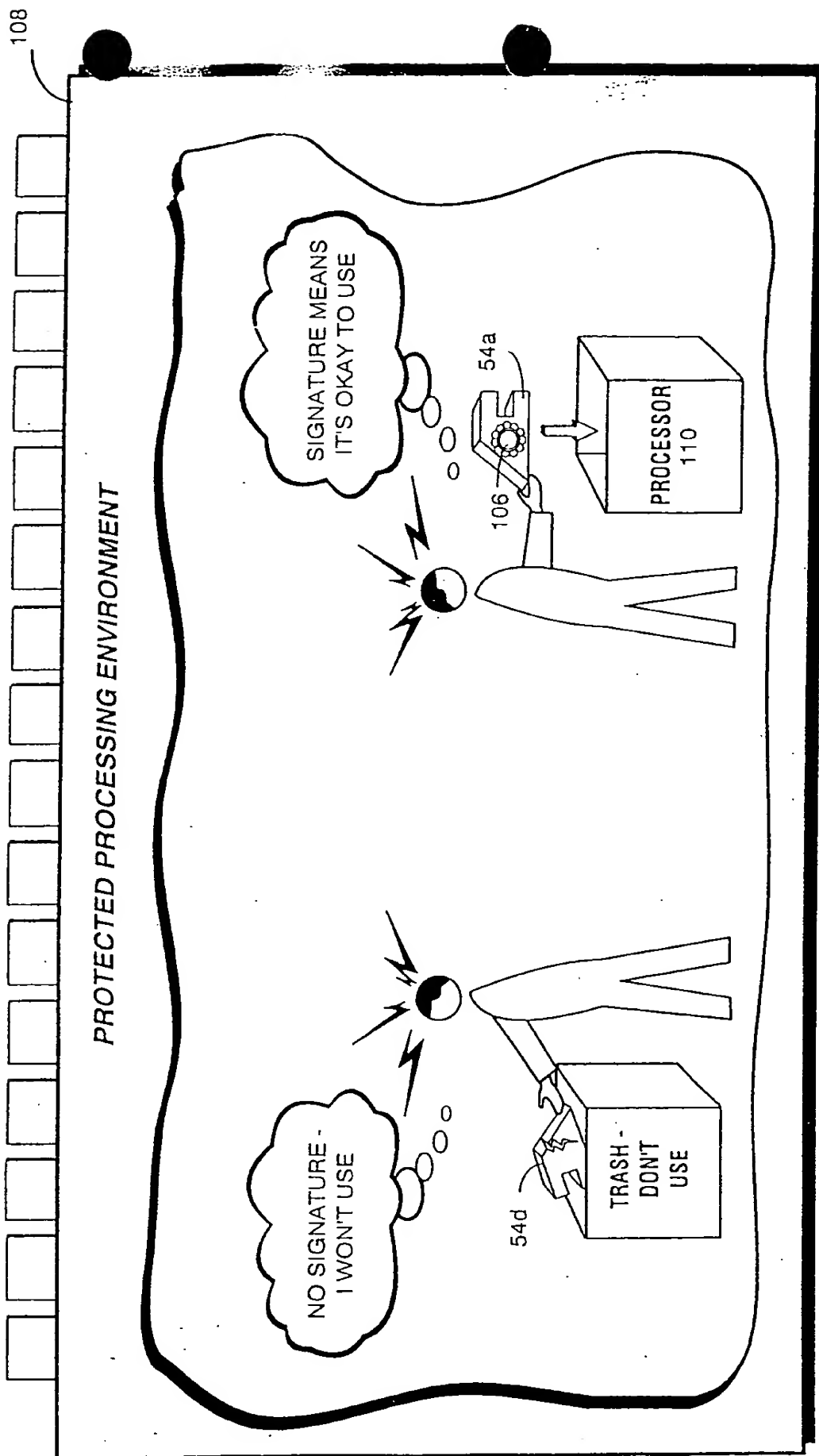


FIG. 3 Before Protected Processing Environment Uses A Load Module, It Checks To See If Load Module Has Been Verified



008270" 26982960

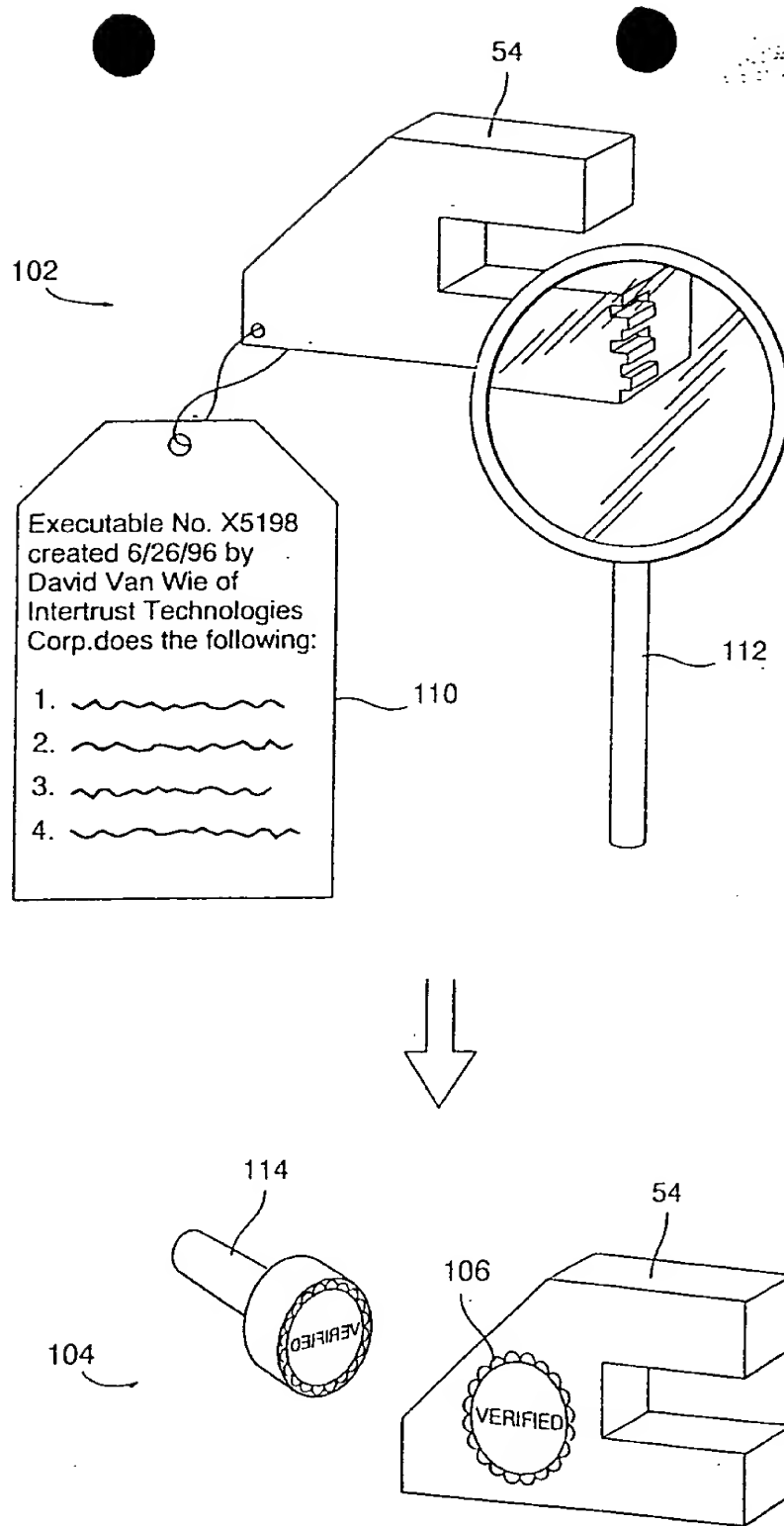


FIG. 4

**Certifying Load Module by
Checking it Against its Documentation**

008270" 26982960

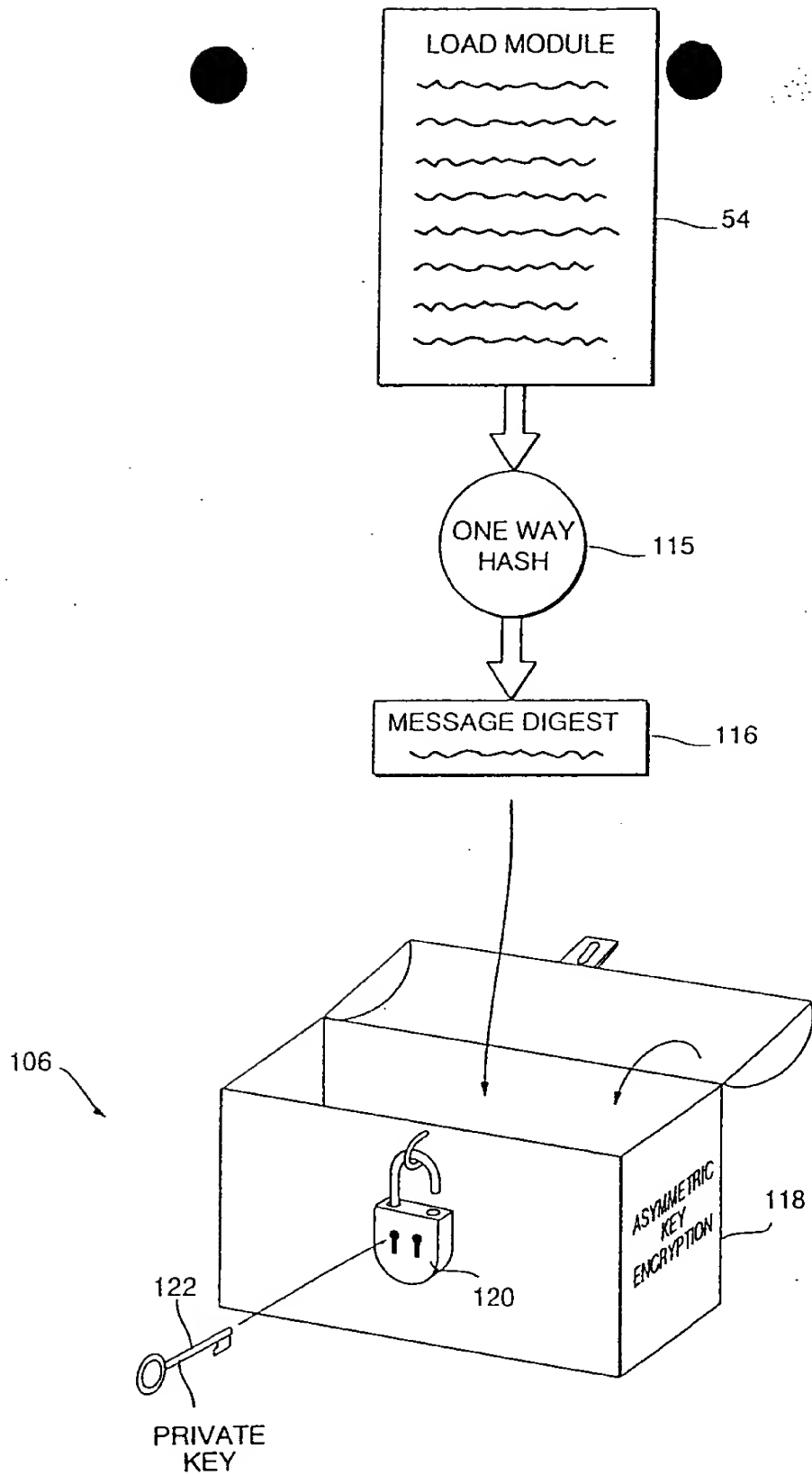


FIG. 5
**Creating a Certifying
Digital Signature**

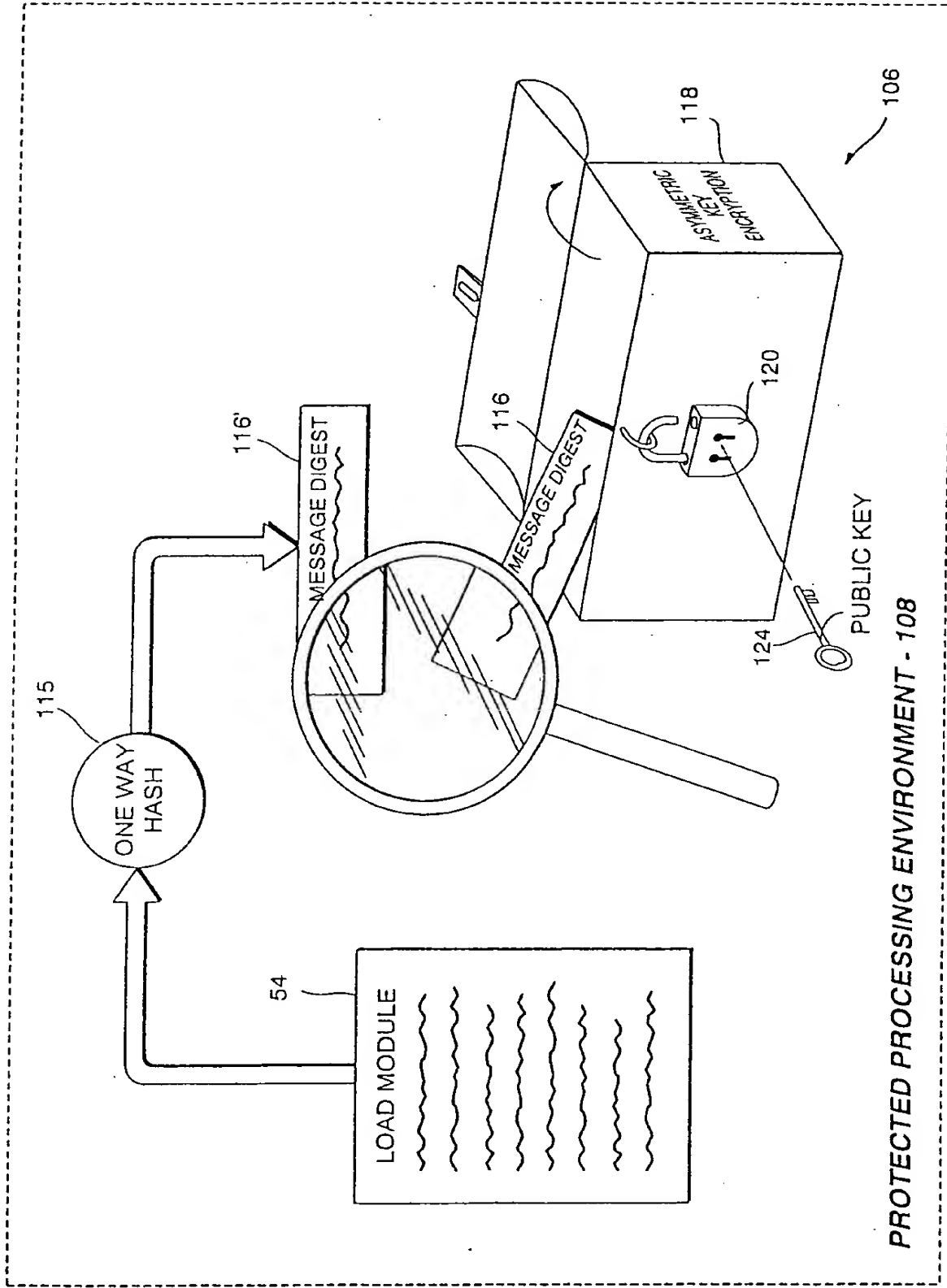


FIG. 6 Authenticating a Digital Signature

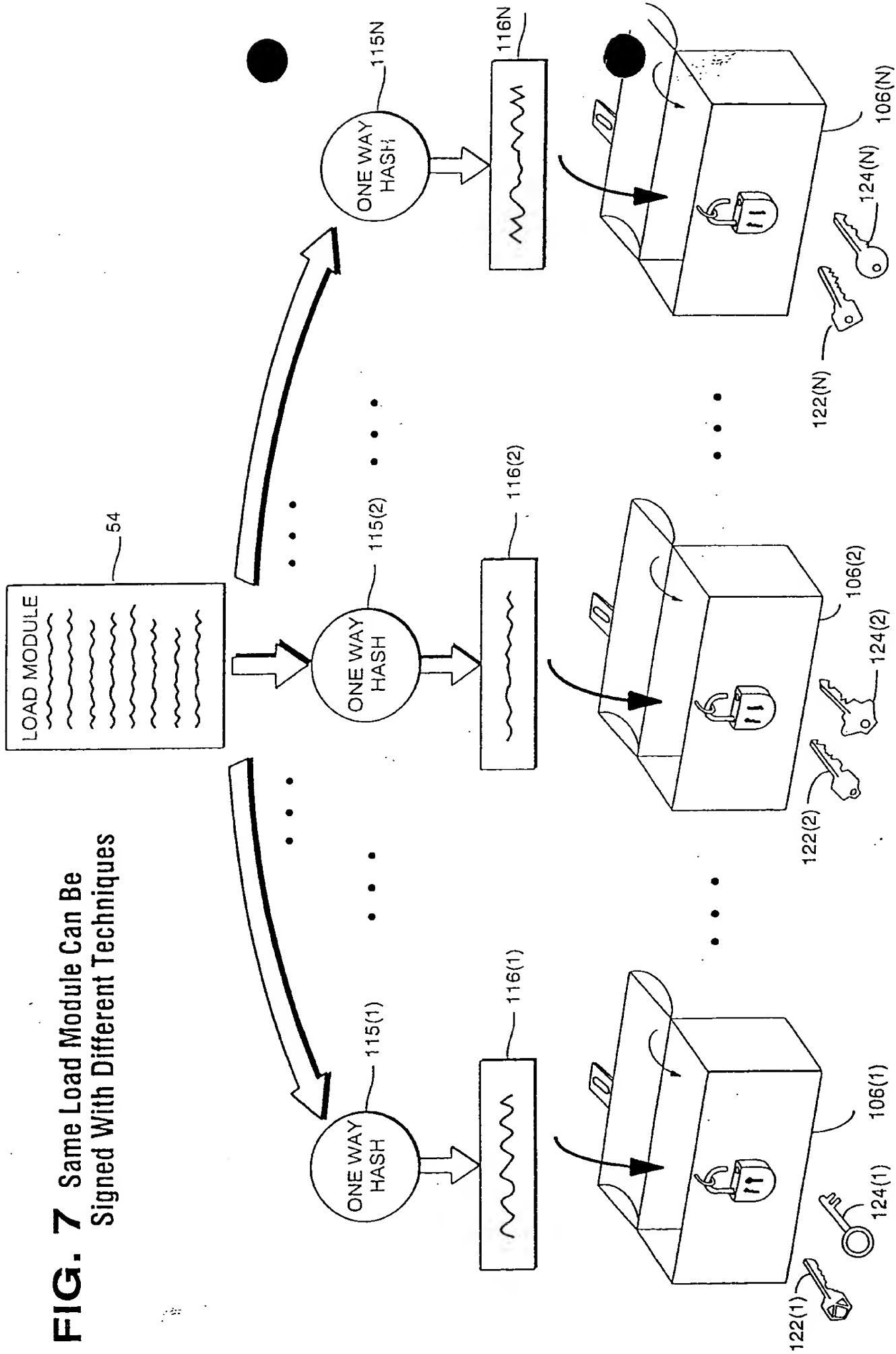


FIG. 7 Same Load Module Can Be Signed With Different Techniques

00000000000000000000000000000000

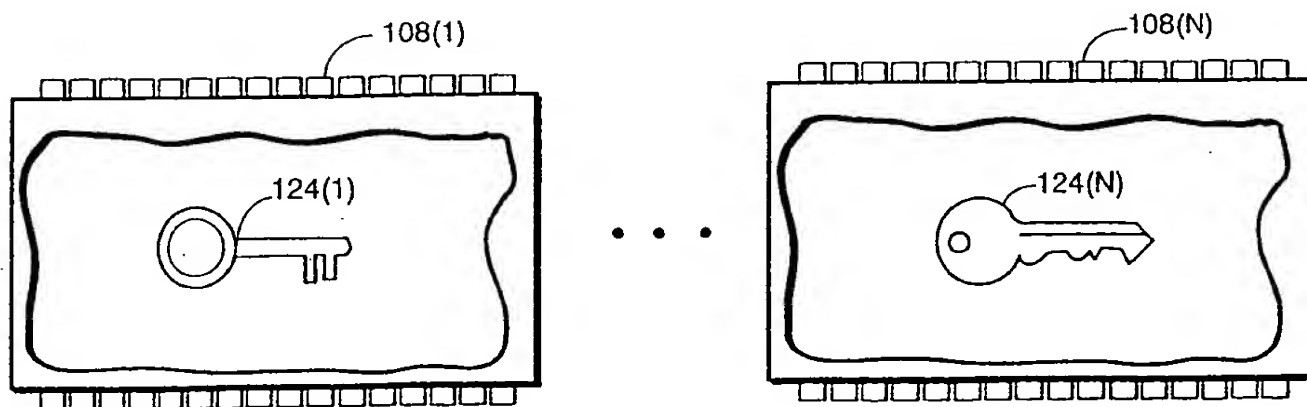
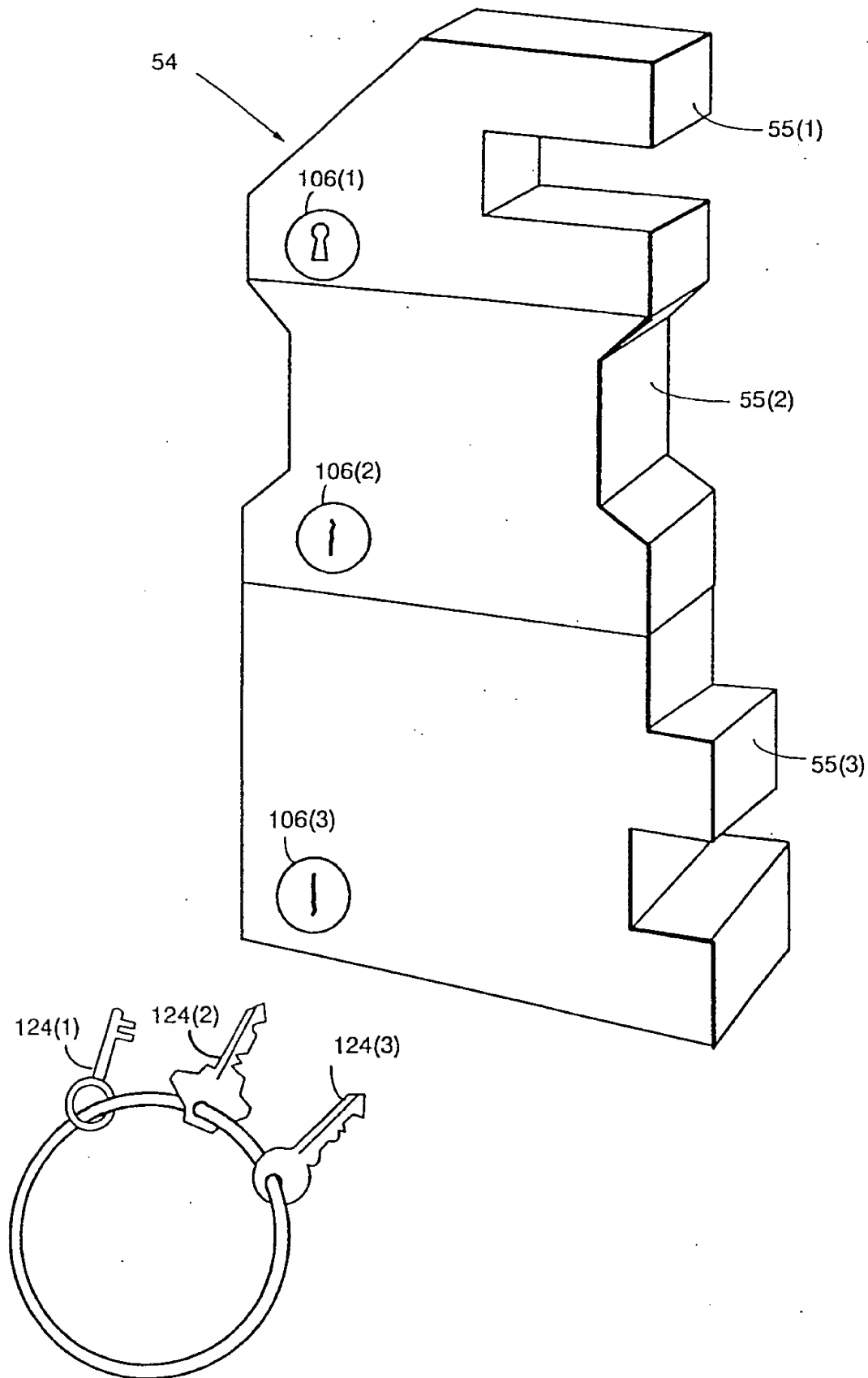


FIG. 9 Load Module Can Have Several Independently Signed Portions



008220" 26982960

FIG. 10A Assurance Level I
Software-Based
Protected Processing Environment

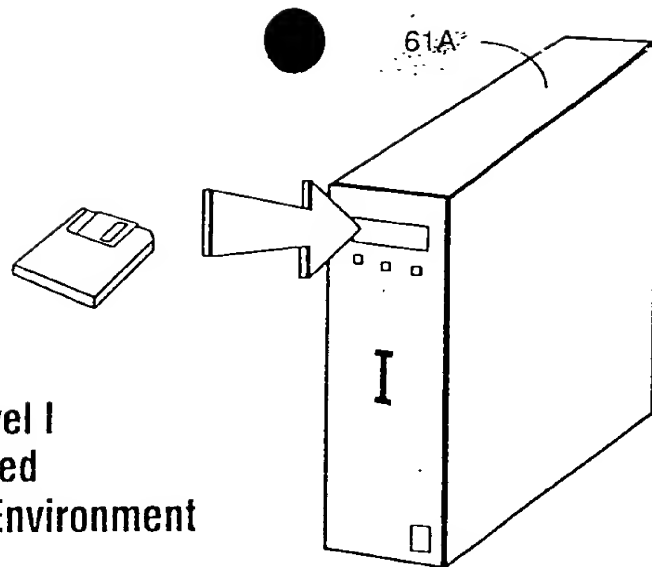


FIG. 10B Assurance Level II
Software and Hardware-Based
Protected Processing Environment

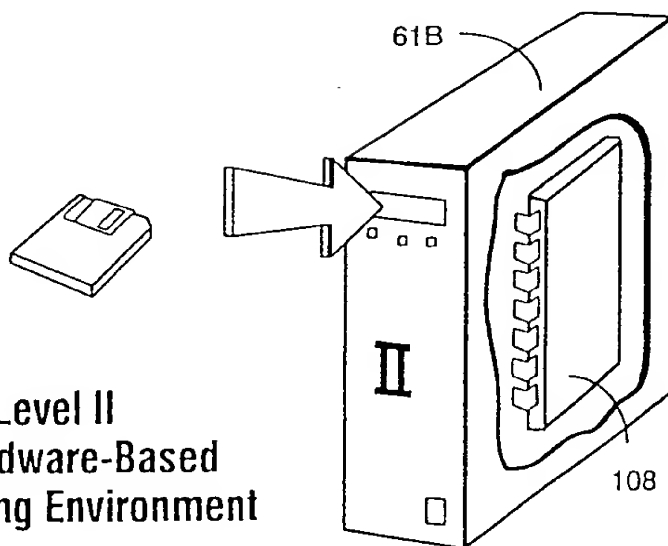
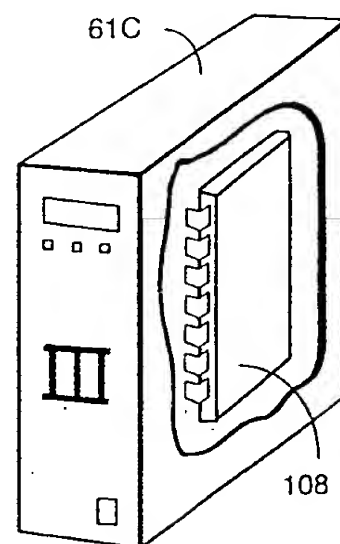


FIG. 10C Assurance Level III
Hardware-Based
Protected Processing Environment



008270-25982960

FIG. 1A Level I
Digital Signature

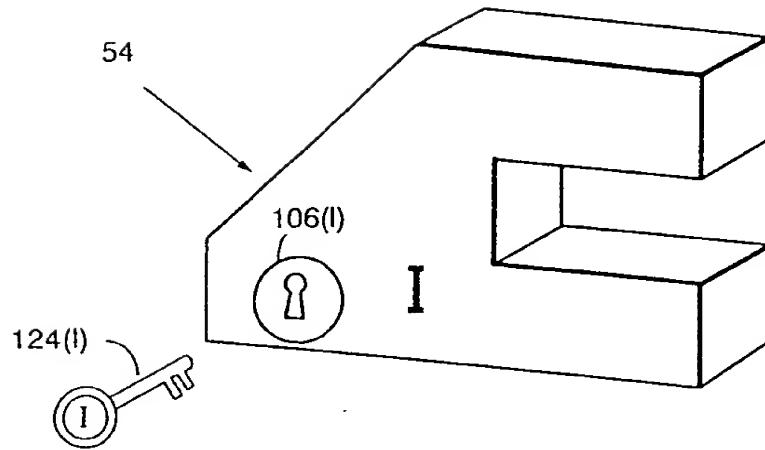


FIG. 11B Level II
Digital Signature

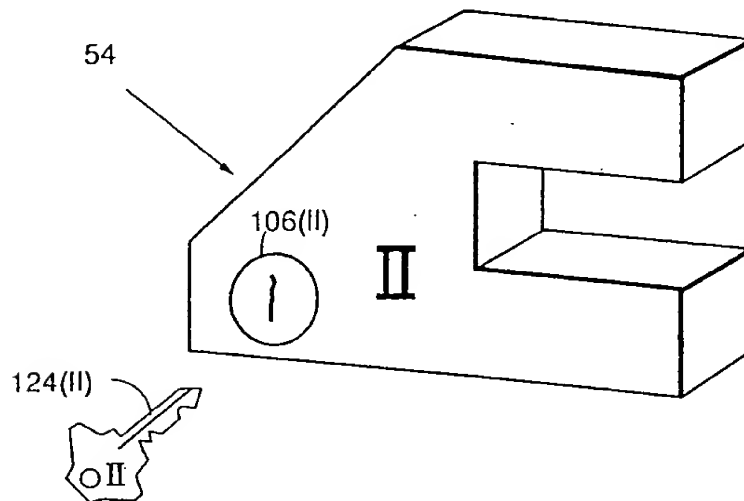
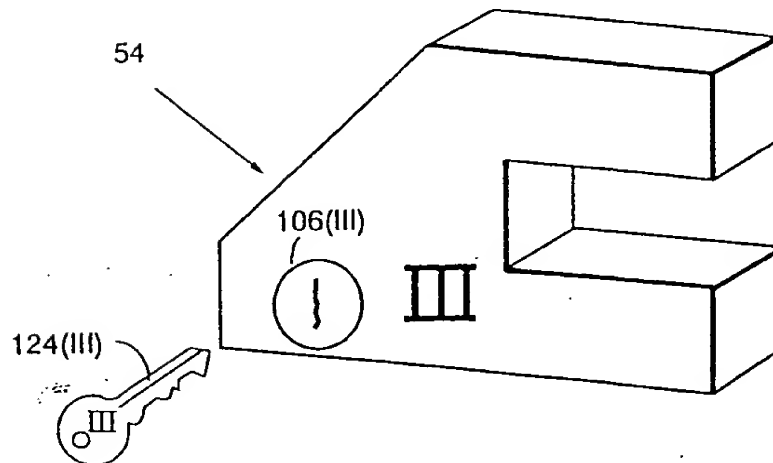


FIG. 11C Level III
Digital Signature



008220" 26982960

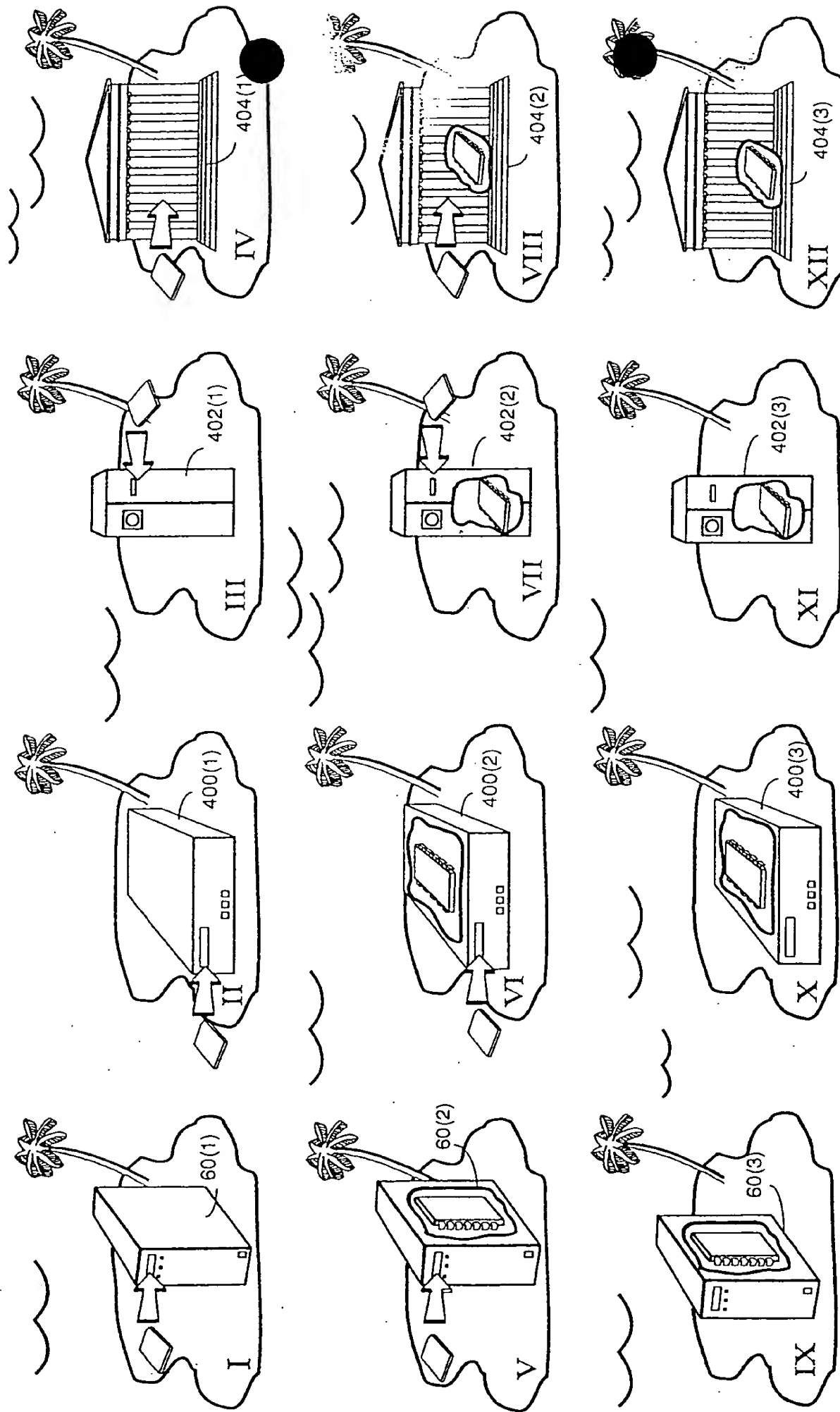


FIG. 12 Using Digital Signatures For Compartmentalizing Different Assurance Levels

FIG. 13 Multiple Assurance Levels

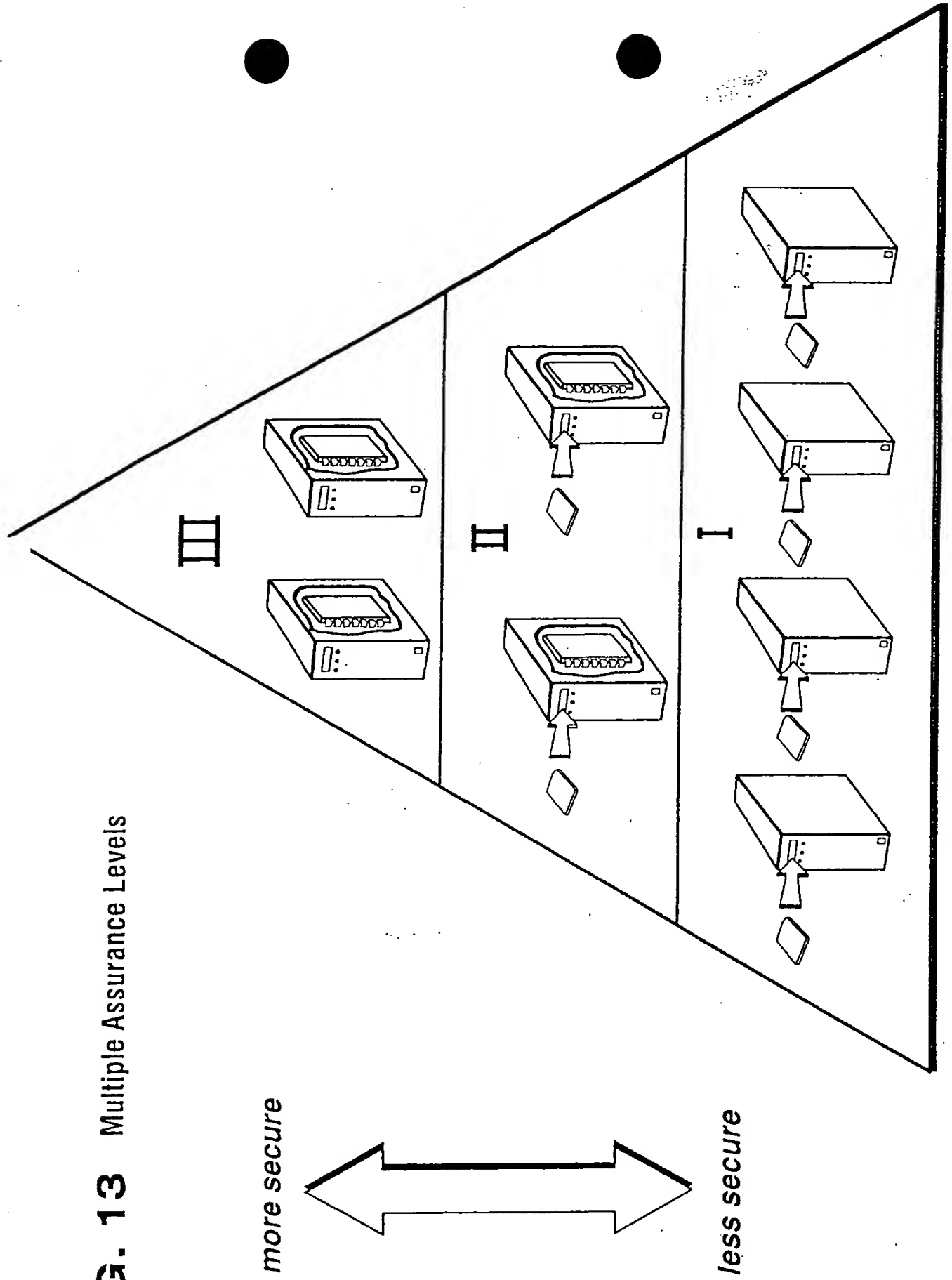
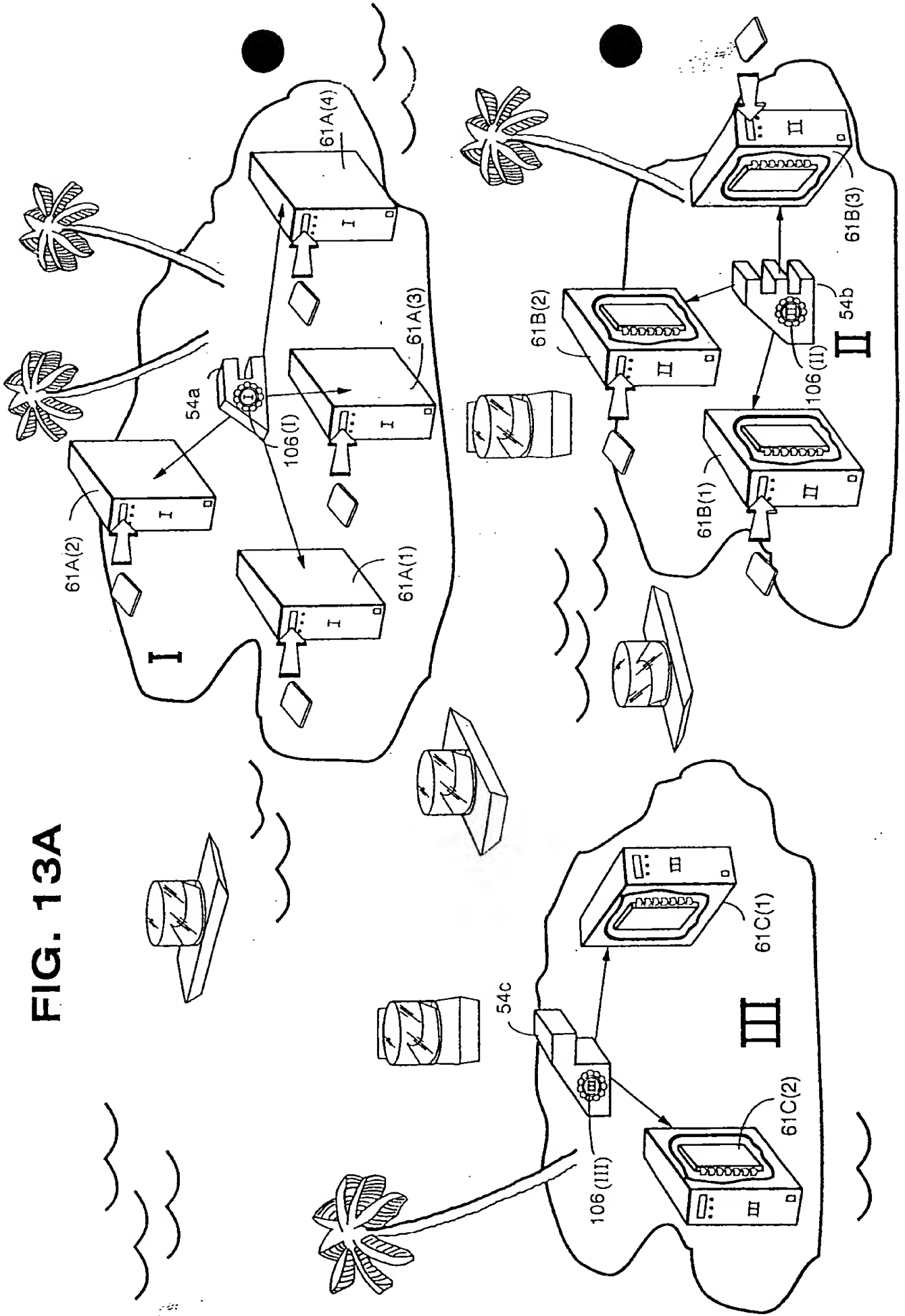
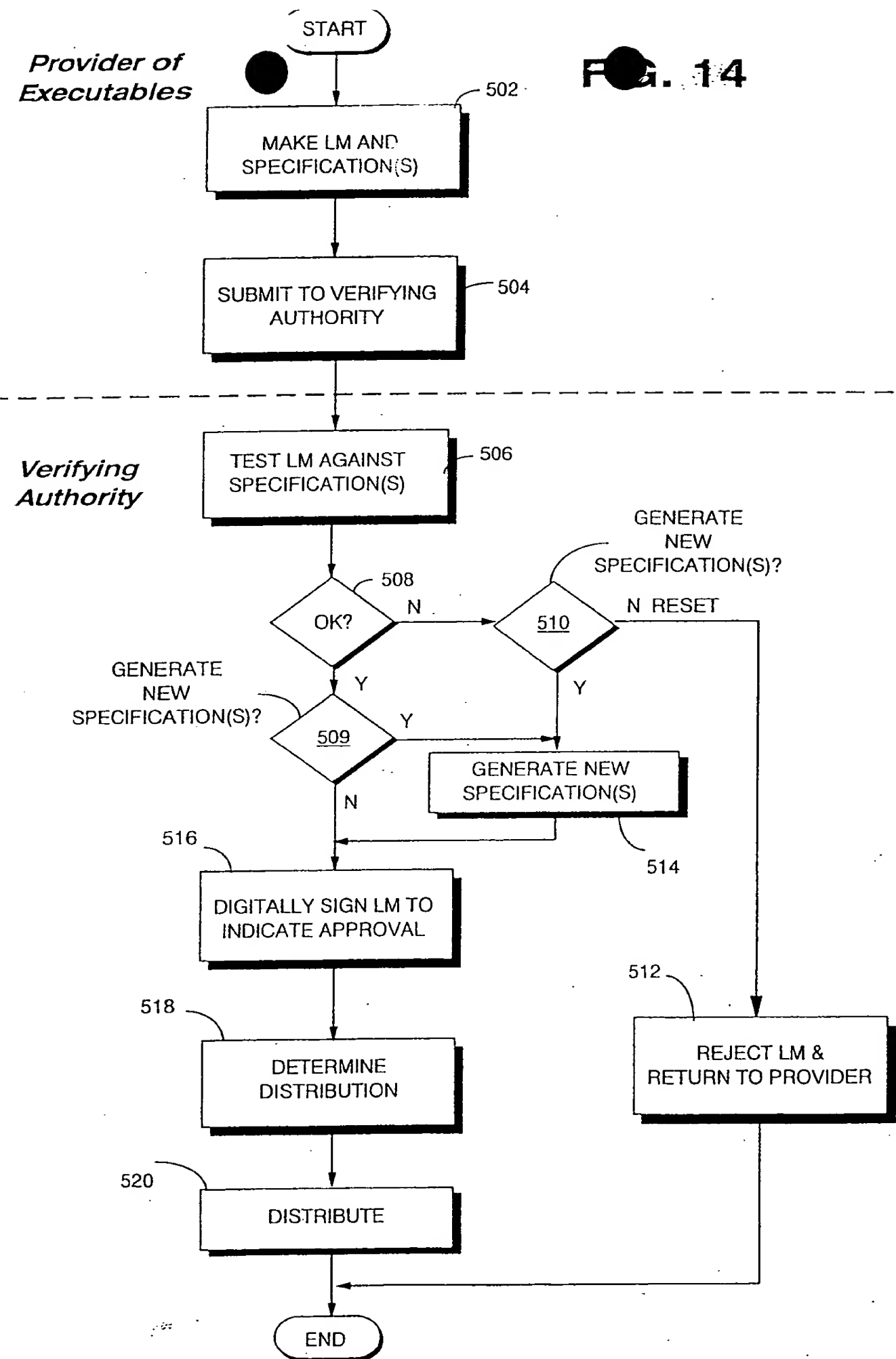


FIG. 13A





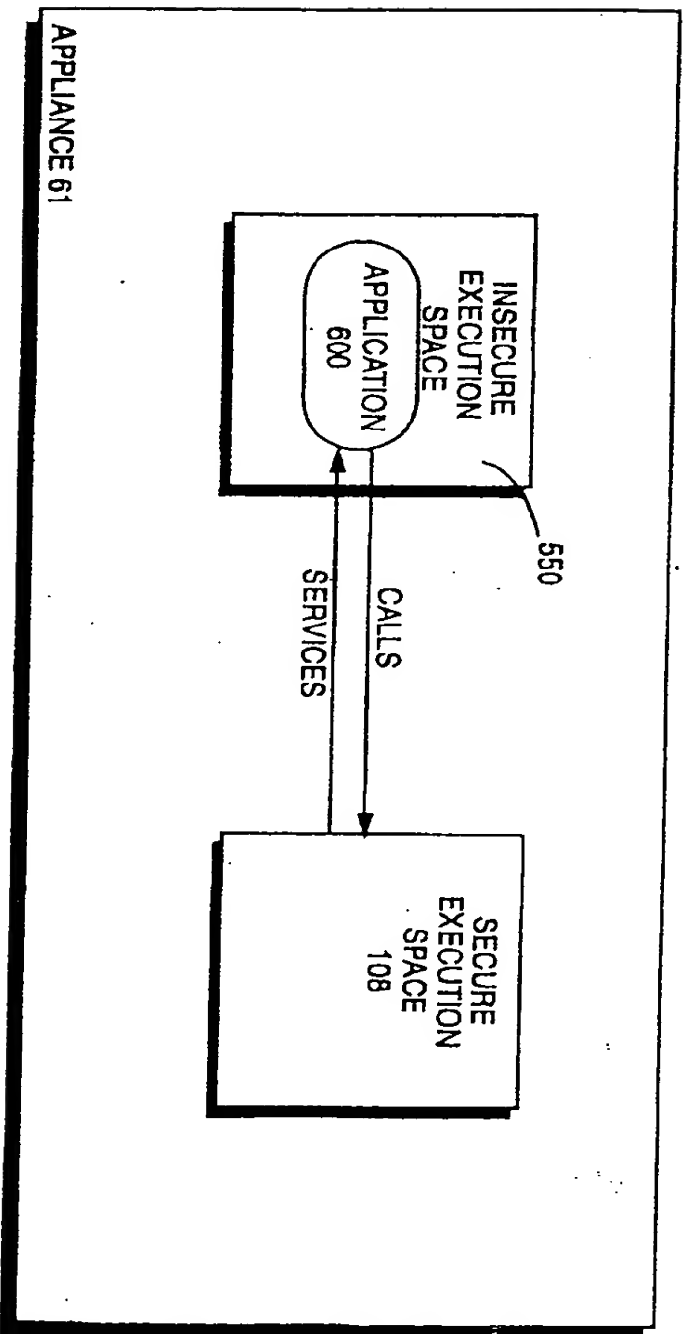


FIG. 15 EXAMPLE APPLIANCE EXECUTING APPLICATION PROGRAM IN INSECURE EXECUTION SPACE

09628592.072800

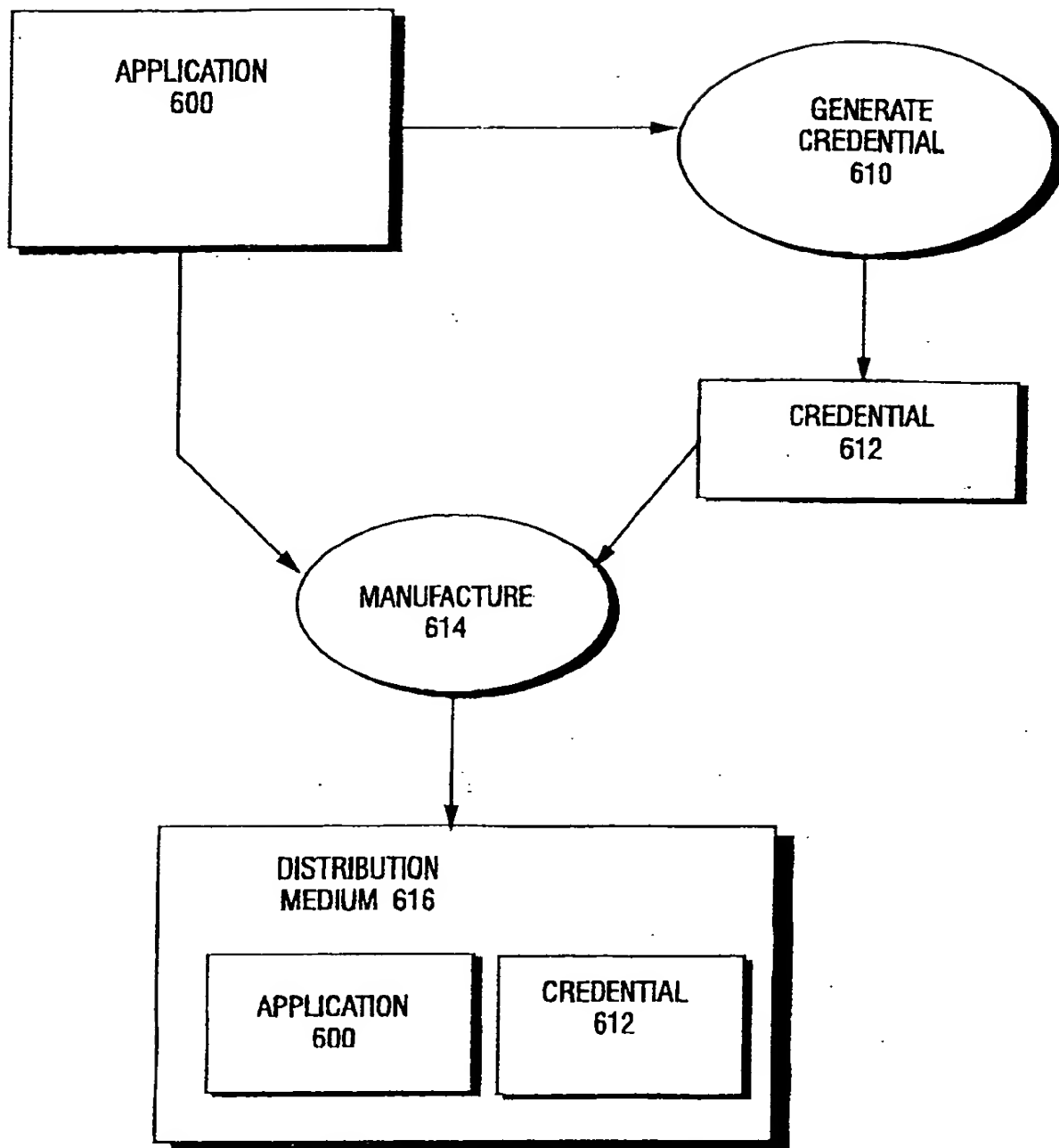


FIG. 16 EXAMPLE APPLICATION CERTIFICATION PROCESS

The diagram illustrates the internal structure of an application, designated as 600. It is enclosed in a large rectangular frame. At the bottom center of the frame, the word "APPLICATION" is written in a large, bold, sans-serif font. Inside the frame, there are five component boxes. In the top-left corner is a box labeled "READ-ONLY COMPONENT 601(N)". In the top-right corner is a box labeled "READ-WRITE COMPONENT 603(1)". In the center of the frame is a box labeled "READ-WRITE COMPONENT 603(N)". To the right of this central box, there is a vertical ellipsis consisting of four small squares. In the bottom-left corner is a box labeled "EXECUTABLE COMPONENT 601(1)". In the bottom-right corner is a box labeled "LIBRARY COMPONENT 601(2)". A curved line extends from the right side of the large frame, pointing to the reference numeral "600".

FIG. 16A EXAMPLE APPLICATION PROGRAM AND COMPONENTS

```
graph TD
    START([START]) --> 700[SELECT APPLICATION COMPONENT PORTION]
    700 --> 702[HASH BYTES IN SELECTED PORTION TO YIELD HASH VALUE]
    702 --> 704[GENERATE HASH BLOCK DESCRIBING EACH CALCULATED PORTION HASH VALUE]
    704 --> 708{REQUIRED QUANTITY OF RANGE HASHES CALCULATED?}
    708 -- N (REPEAT) --> 700
    708 -- Y --> 710[HASH SET OF HASH BLOCKS]
    710 --> 712[DIGITALLY SIGN THE HASH]
    712 --> 714[ENCRYPT SET OF HASH BLOCKS AND DIGITIZE SIGNATURE TO CREATE CREDENTIAL PART]
    714 --> 716[COMBINE CREDENTIAL PARTS TO PRODUCE CREDENTIAL]
    716 --> END([END])
```

FIG. 17

EXAMPLE CREDENTIAL CREATION PROCESS

003270" 26982960

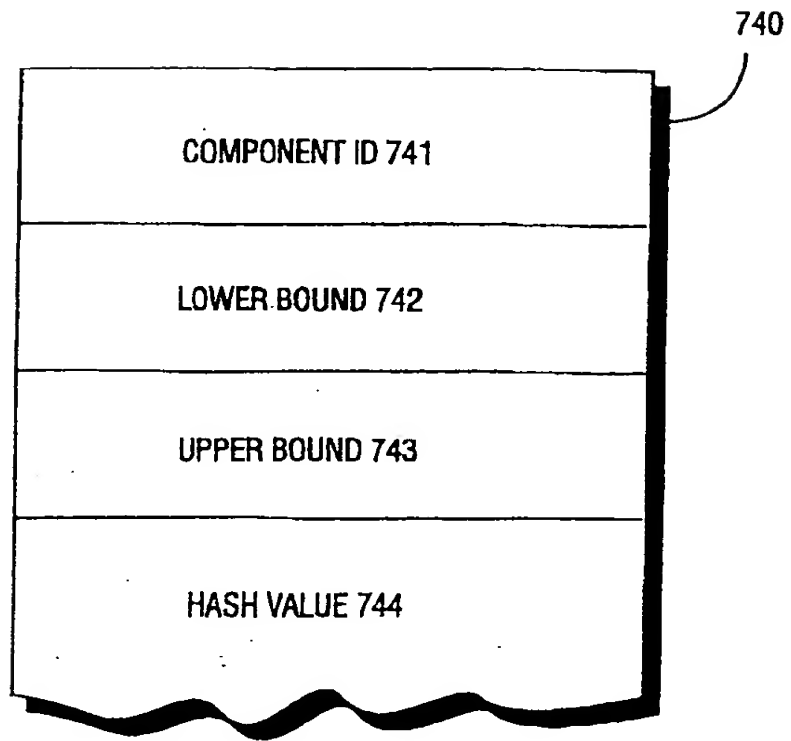


FIG. 18 EXAMPLE HASH BLOCK

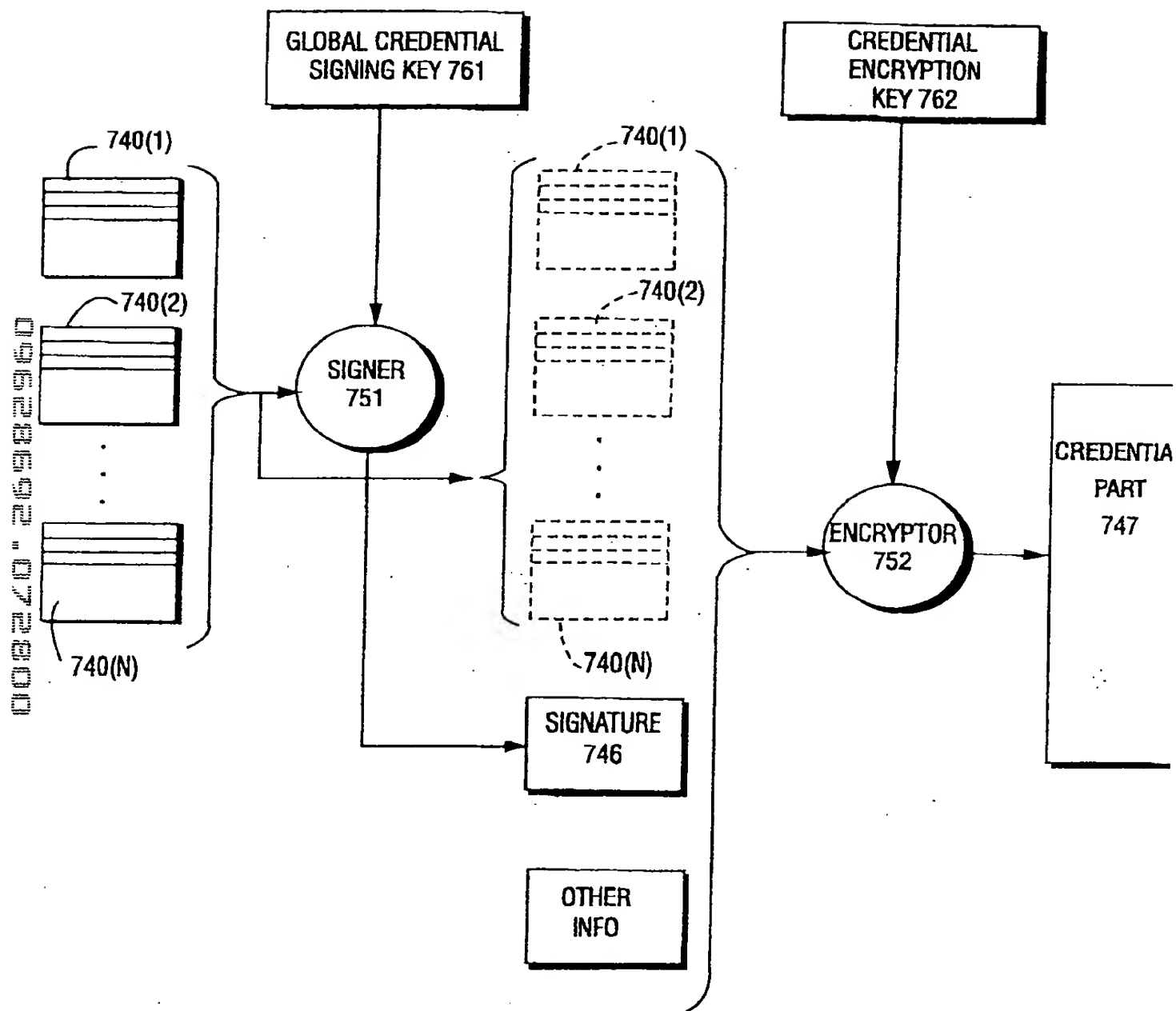


FIG. 19 EXAMPLE CREDENTIAL CREATION

The diagram illustrates a security attack on a validation process. At the top, a large box labeled "APPLICATION 600" contains a smaller box labeled "CRITICAL COMPONENT 804". To the right of the application box, a vertical bracket labeled "VALIDATION RANGES" indicates a sequence of checks. These checks result in "OK" or "FAIL" outcomes, shown as a series of horizontal lines branching from the bracket. Below the application box, an arrow labeled "ATTACKER SUBSTITUTES" points from a box labeled "ATTACKER'S CRITICAL COMPONENT 802" to the "CRITICAL COMPONENT 804" box. A figure of an attacker, labeled "64", is shown holding the "ATTACKER'S CRITICAL COMPONENT 802" box.

45

008270" 26982960

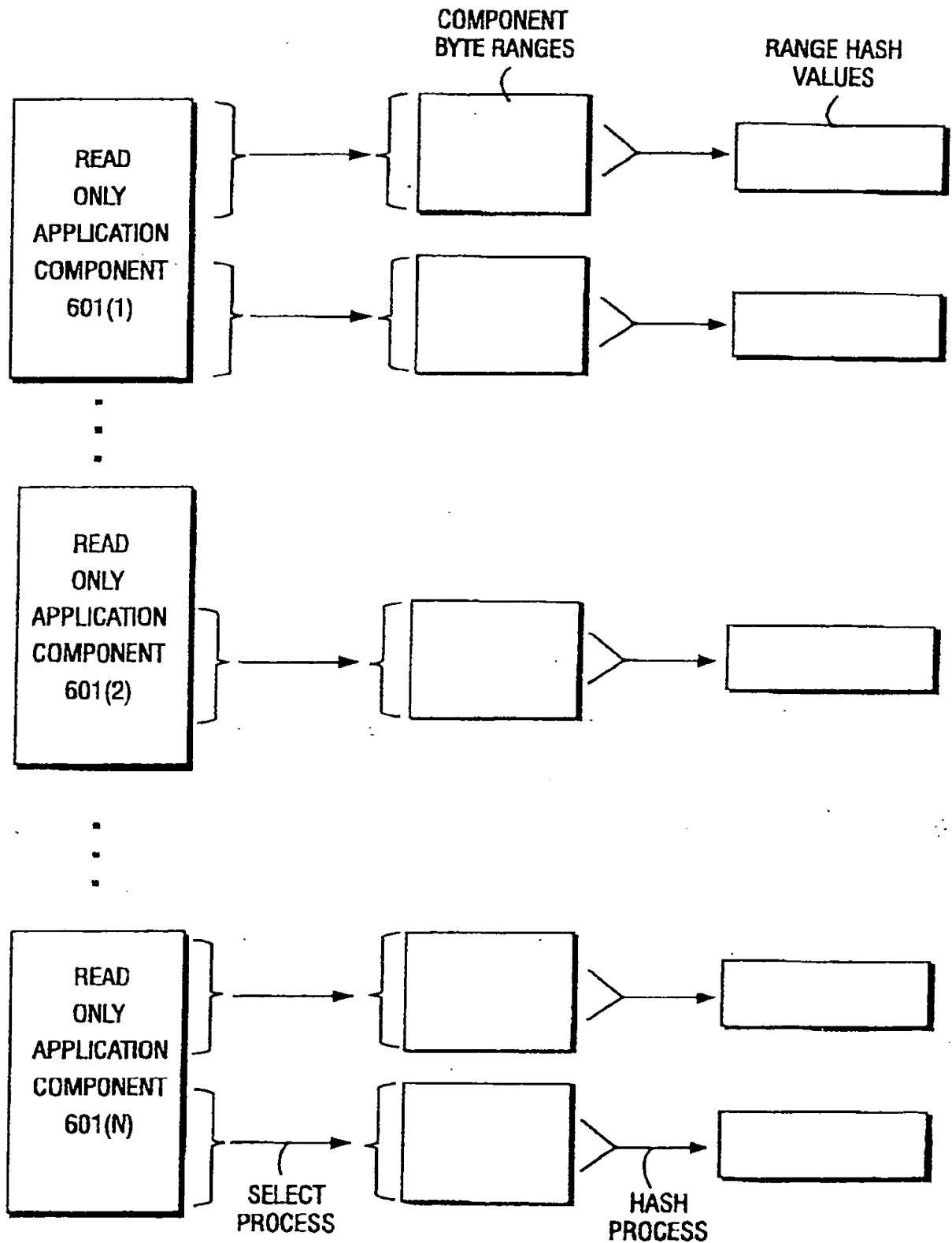


FIG. 20A EXAMPLE NON-OVERLAPPING HASH RANGES

008220" 26982960

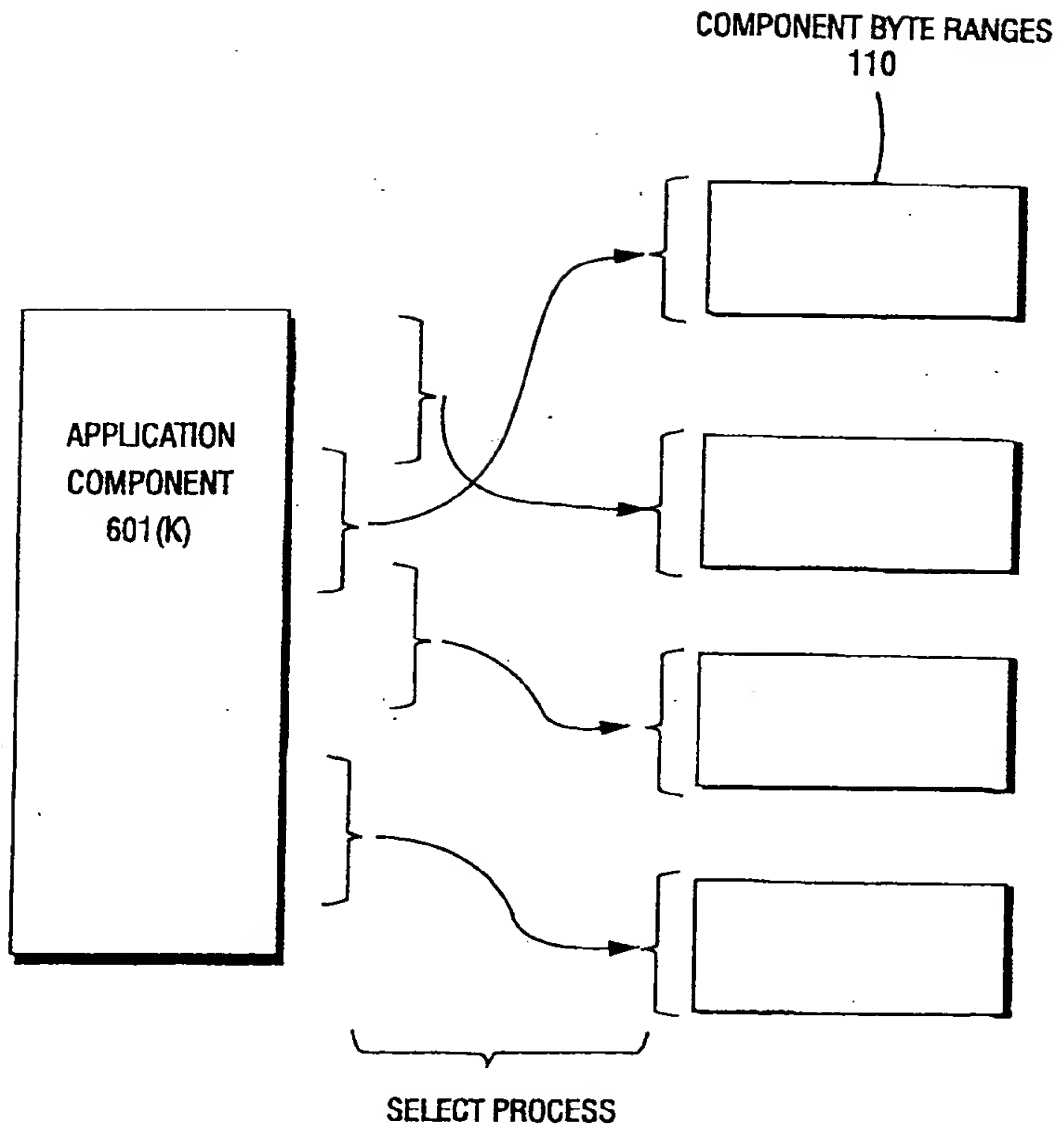


FIG. 20B EXAMPLE OF OVERLAPPING HASH RANGES

00000000000000000000000000000000

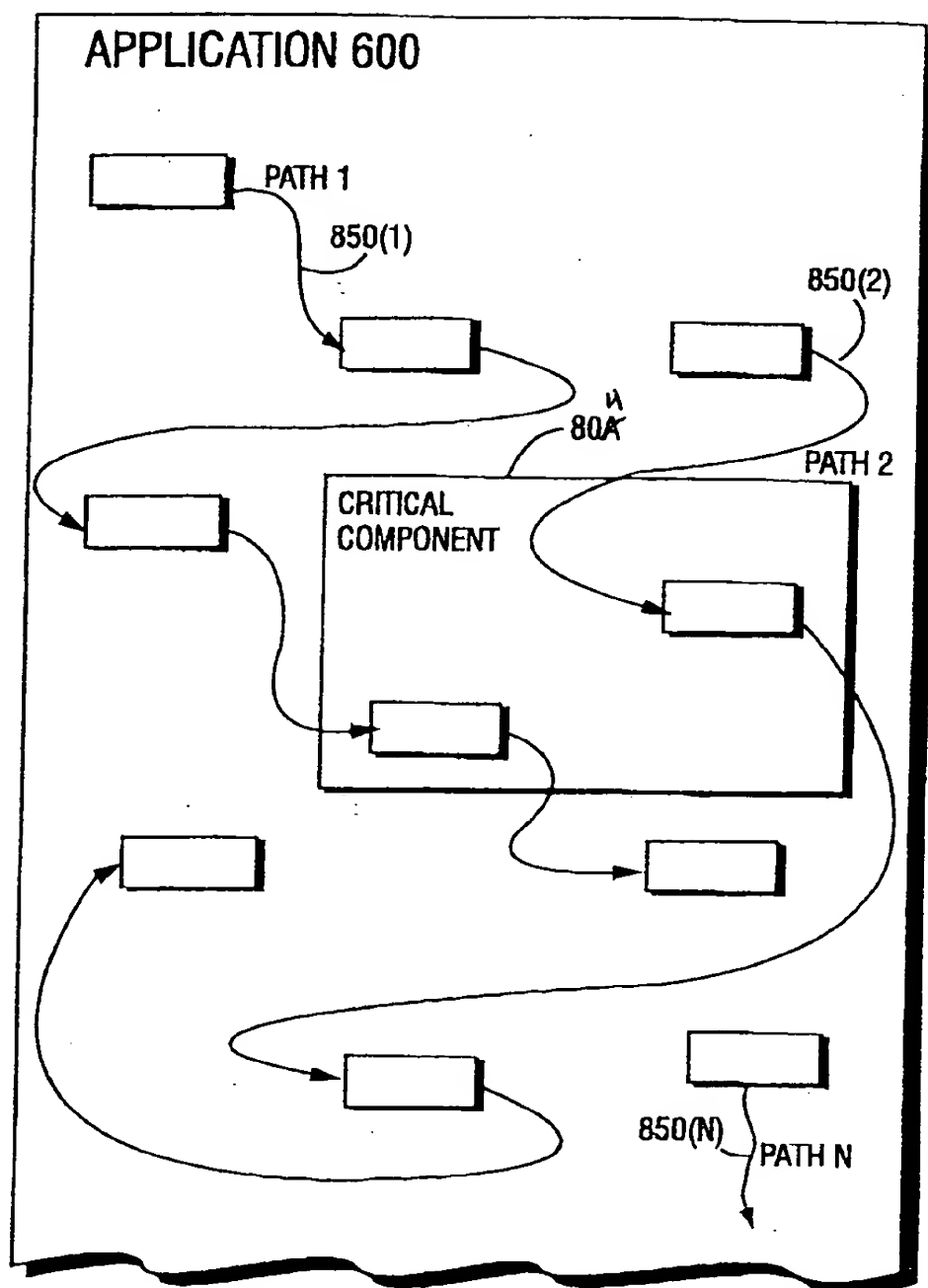


FIG. 20C PSEUDO-RANDOM VALIDATION PATHS IN APPLICATION

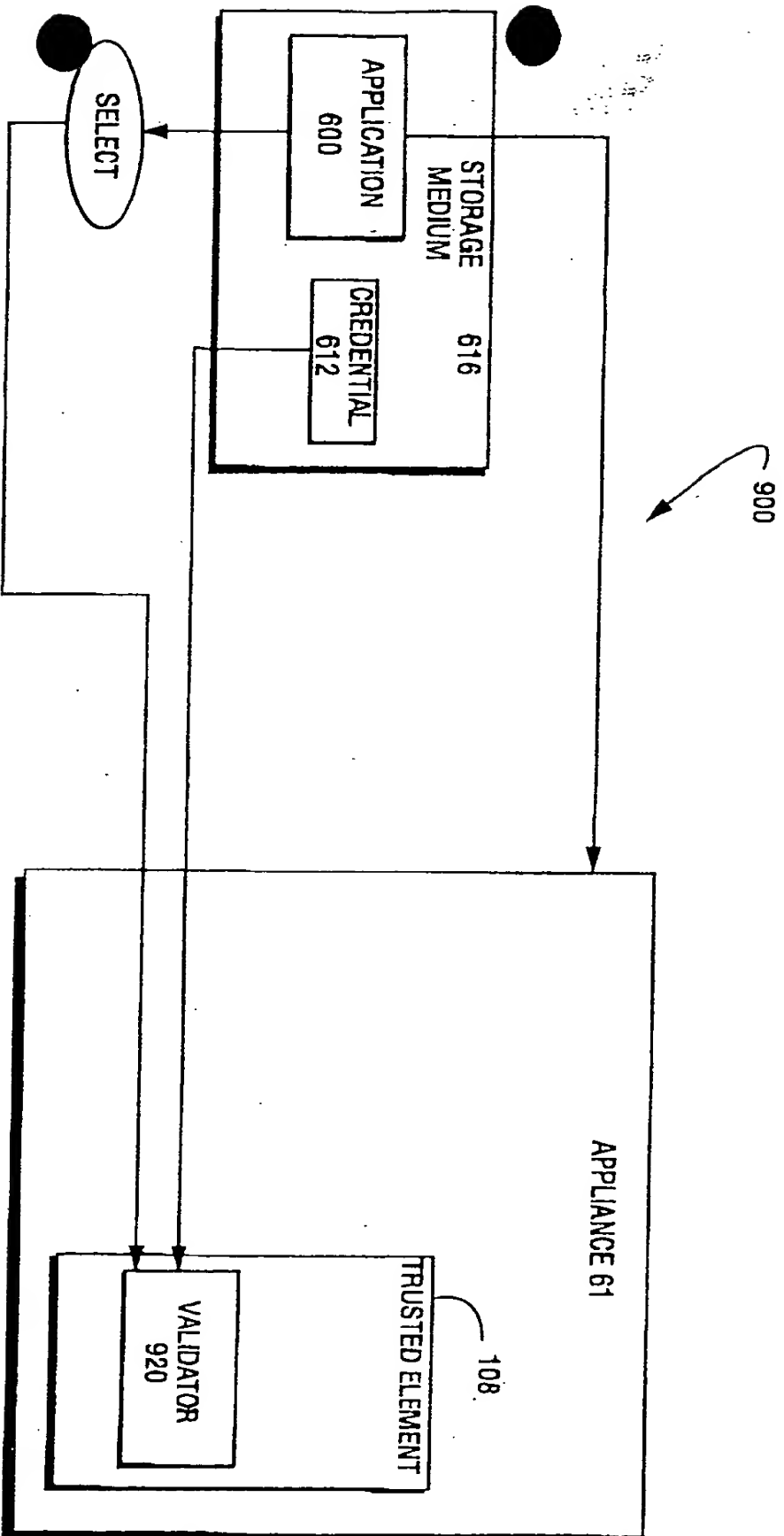
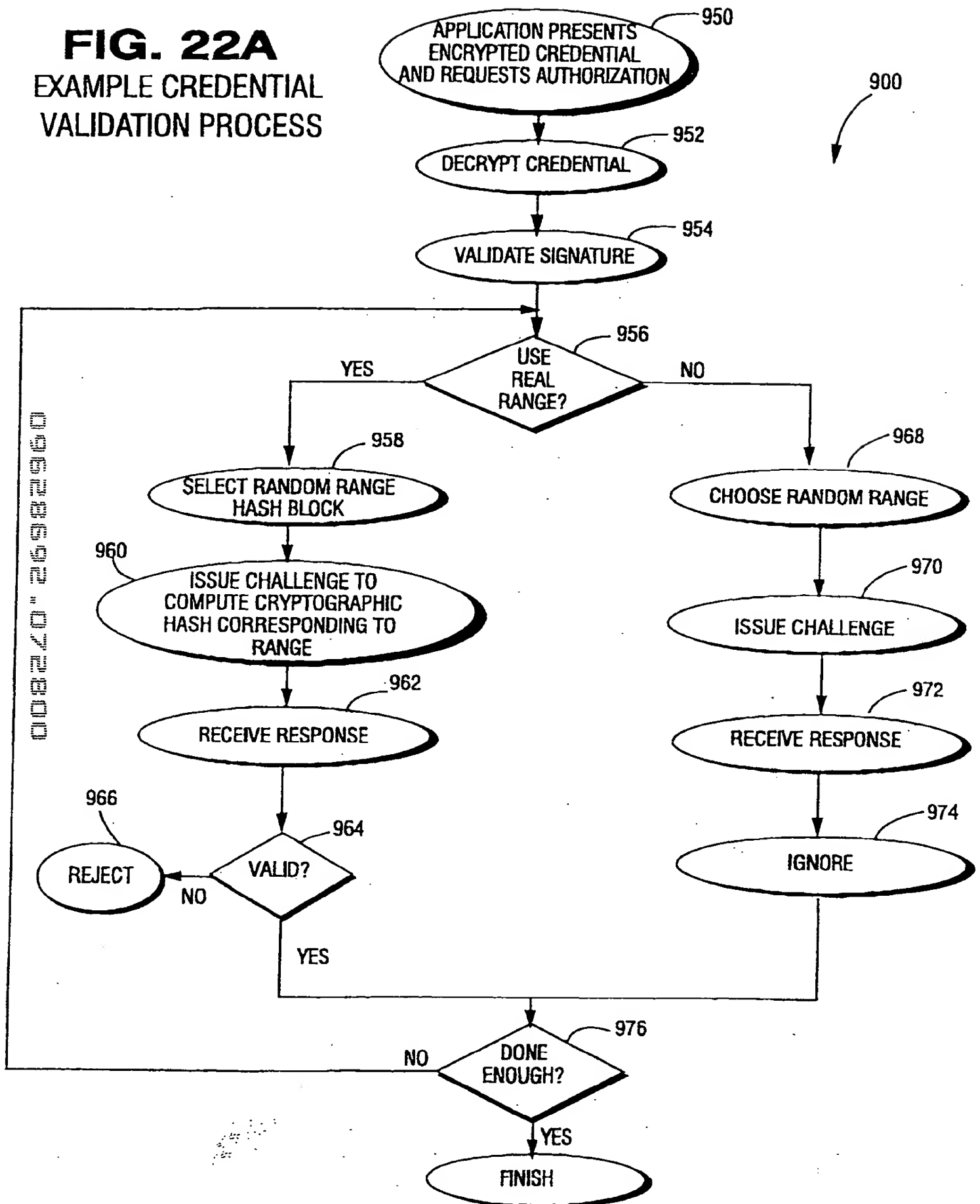


FIG. 21 EXAMPLE CREDENTIAL VALIDATION PROCESS

FIG. 22A
EXAMPLE CREDENTIAL
VALIDATION PROCESS



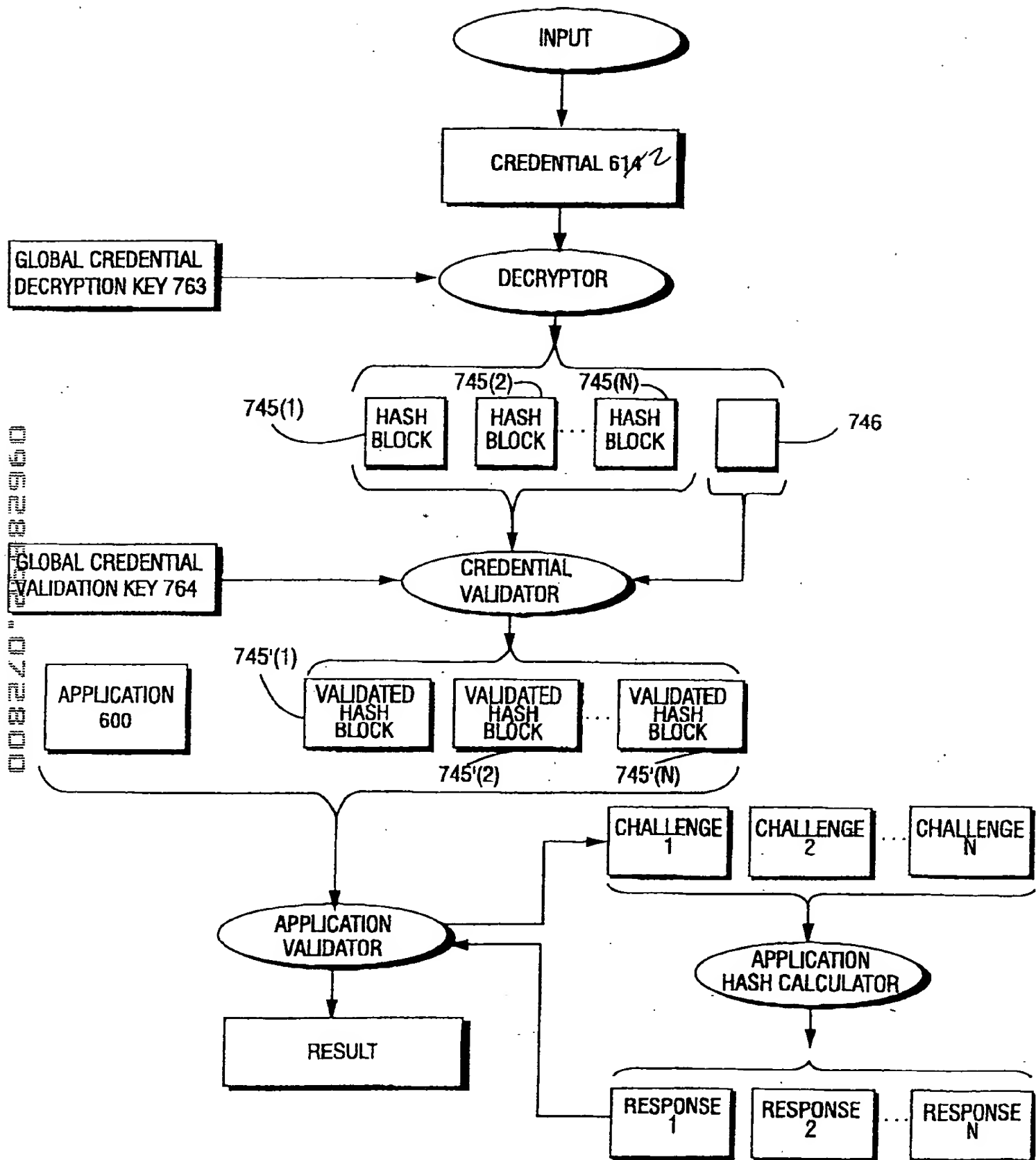


FIG. 22B EXAMPLE CREDENTIAL VALIDATION